```
> restart;
  with(PDEtools):
  with(LinearAlgebra):
```

Introduction to partial differential equations in Maple

The purpose of this worksheet it to give a (very brief) introduction to partial differential equations (PDEs) and Maple's capabilities to solve ther both analytically and numerically. We will be making extensive use of the routines contained in the **PDEtools** and **LinearAlgebra** packages, which have been loaded after the **restart** command above.

Classification of linear partial differential equations of order ≤ 2

Let us begin by writing down the most general linear PDE in N dimensions with differential order less than 2 and constant coefficients. Thi will be defined by an $N \times N$ symmetric constant matrix $\boldsymbol{A}$, an N-dimensional constant vector $\boldsymbol{B}$, and a scalar c:

```
> N := 2;
  vars := seq(x[k],k=1..N);
  A := Matrix(N,symbol=a,shape=symmetric);
  B := Vector(N,symbol=b);
  D2 := convert(Matrix([seq([seq(D[i,j](phi)(vars),i=1..N)],j=1..N)]),diff);
  D1 := convert(Vector[row]([seq(D[i](phi)(vars),i=1..N)]),diff);
  pde := Trace(A.D2) + D1.B + c*phi(vars);
```

$$N := 2$$

$$vars := x_1, x_2$$

$$A := \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{1,2} & a_{2,2} \end{bmatrix}$$

$$B := \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$D2 := \begin{bmatrix} \frac{\partial^2}{\partial x_1^{\,2}} \phi(x_1, x_2) & \frac{\partial^2}{\partial x_2 \, \partial x_1} \phi(x_1, x_2) \\ \frac{\partial^2}{\partial x_2 \, \partial x_1} \phi(x_1, x_2) & \frac{\partial^2}{\partial x_2^{\,2}} \phi(x_1, x_2) \end{bmatrix}$$

$$D1 := \left[\frac{\partial}{\partial x_1} \phi(x_1, x_2) \quad \frac{\partial}{\partial x_2} \phi(x_1, x_2) \right]$$

$$pde := a_{1,1} \left(\frac{\partial^2}{\partial x_1^{\,2}} \phi(x_1, x_2) \right) + 2\, a_{1,2} \left(\frac{\partial^2}{\partial x_2 \, \partial x_1} \phi(x_1, x_2) \right) + a_{2,2} \left(\frac{\partial^2}{\partial x_2^{\,2}} \phi(x_1, x_2) \right) + b_1 \left(\frac{\partial}{\partial x_1} \phi(x_1, x_2) \right) + b_2 \left(\frac{\partial}{\partial x_2} \phi(x_1, x_2) \right) + c\, \phi(x_1, x_2) \tag{1.1}$$

We see in **pde** that all possible derivatives of the unknown function ϕ are represented in this expression with a unique constant coefficient. We can perform a linear transformation the independent variables $\boldsymbol{x} = (x_1, x_2)$ to bring this into a canonical form. The transformation will defined by $\boldsymbol{x} = \Lambda \boldsymbol{y}$ where Λ is an orthogonal matrix that diagonalizes $\boldsymbol{A}$. To construct such a matrix, we use the **Eigenvectors(A)** command, which returns two objects: the first **lambda** is a vector of the eigenvalues of $\boldsymbol{A}$, the second is a matrix $\boldsymbol{P}$ whose columns are eigenvectors of $\boldsymbol{A}$ (N.B. the following code will be extremely processor intensive for $N \geq 3$):

```
> gnat := Eigenvectors(A):
  lambda := gnat[1];
  P := gnat[2];
```

$$\lambda := \begin{bmatrix} \frac{1}{2} a_{2,2} + \frac{1}{2} a_{1,1} + \frac{1}{2} \sqrt{a_{2,2}^2 - 2\, a_{2,2}\, a_{1,1} + a_{1,1}^2 + 4\, a_{1,2}^2} \\ \frac{1}{2} a_{2,2} + \frac{1}{2} a_{1,1} - \frac{1}{2} \sqrt{a_{2,2}^2 - 2\, a_{2,2}\, a_{1,1} + a_{1,1}^2 + 4\, a_{1,2}^2} \end{bmatrix}$$

$$P := \left[\begin{bmatrix} \frac{a_{1,2}}{\frac{1}{2} a_{2,2} - \frac{1}{2} a_{1,1} + \frac{1}{2} \sqrt{a_{2,2}^2 - 2\, a_{2,2}\, a_{1,1} + a_{1,1}^2 + 4\, a_{1,2}^2}} \\ \frac{a_{1,2}}{\frac{1}{2} a_{2,2} - \frac{1}{2} a_{1,1} - \frac{1}{2} \sqrt{a_{2,2}^2 - 2\, a_{2,2}\, a_{1,1} + a_{1,1}^2 + 4\, a_{1,2}^2}} \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right] \tag{1.2}$$

The eigenvectors contained in the columns of $\boldsymbol{P}$ are unnormalized; the following code constructs the Λ matrix with columns corresponding t

the normalized eigenvectors of $\boldsymbol{A}$.

```
> for i from 1 to 2 do:
    v[i] := Column(P,i);
    N := sqrt(v[i]^%T.v[i]);
    v[i] := v[i]/N;
  od:

  Lambda := simplify(Matrix([v[1],v[2]]),symbolic);
```

$$\Lambda := \left[\left[\left(2\,a_{1,2}\right)\Big/\left(2\,a_{2,2}^2-4\,a_{2,2}\,a_{1,1}+2\,a_{2,2}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}+2\,a_{1,1}^2\right.\right.\right.$$

$$\left.\left.\left.-2\,a_{1,1}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}+8\,a_{1,2}^2\right)^{1/2},-\left(a_{1,2}\sqrt{2}\right)\Big/\right.$$

$$\left.\left(a_{2,2}^2-2\,a_{2,2}\,a_{1,1}-a_{2,2}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}+a_{1,1}^2\right.\right.$$

$$\left.\left.+a_{1,1}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}+4\,a_{1,2}^2\right)^{1/2}\right],$$

$$\left[\left(a_{2,2}-a_{1,1}+\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}\right)\Big/\right.$$

$$\left(2\,a_{2,2}^2-4\,a_{2,2}\,a_{1,1}+2\,a_{2,2}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}+2\,a_{1,1}^2\right.$$

$$\left.\left.-2\,a_{1,1}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}+8\,a_{1,2}^2\right)^{1/2},\frac{1}{2}\left(\sqrt{2}\left(-a_{2,2}+a_{1,1}\right.\right.\right.$$

(1.3

$$+\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}\,\Big)\Big)\Big/$$

$$\Big(a_{2,2}^2-2\,a_{2,2}\,a_{1,1}-a_{2,2}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}+a_{1,1}^2$$

$$+a_{1,1}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}+4\,a_{1,2}^2\Big)^{1/2}\Big]\Big]$$

We now confirm that $\Lambda^T A\Lambda$ is indeed a diagonal matrix whose entries are the eigenvalues of A. Note the use of `Lambda^%T` to calculate the transpose and `DiagonalMatrix(lambda)` to construct a diagonal matrix whose diagonal entries are given by the vector `lambda`:

```
> simplify(Lambda^%T.A.Lambda-DiagonalMatrix(lambda));
```

$$\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix} \qquad \textbf{(1.4}$$

Explicit equations for the transformation $\boldsymbol{x} = \Lambda\boldsymbol{y}$ can be obtained by using the `GenerateEquations` command (we've supressed the output, it is pretty long):

```
> X := Vector([vars]):
  tr := solve(GenerateEquations(Lambda,[y[1],y[2]],X),{vars}):
```

We can use the `dchange` command to transform `pde` to the (y_1, y_2) coordinate system via the transformation `tr`:

```
> pde := collect(convert(simplify(dchange(tr,pde,[y[1],y[2]])),D),phi,factor);
```

$$pde := c\,\phi\left(y_1, y_2, a_{1,1}, a_{1,2}, a_{2,2}\right) \qquad \textbf{(1.5}$$

$$+\frac{1}{4}\,\frac{1}{a_{1,2}\left(a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2\right)}\Big(\Big(2\,a_{2,2}^2-4\,a_{2,2}\,a_{1,1}$$

$$+2\,a_{2,2}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}+2\,a_{1,1}^2-2\,a_{1,1}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}+8\,a_{1,2}^2\Big)$$

$$^{1/2}\Big(4\,a_{1,2}^2\,b_1+\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}\;a_{1,1}\,b_1-\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}\;a_{2,2}\,b_1$$

$$-2\,a_{1,1}\,a_{2,2}\,b_1+2\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}\;a_{1,2}\,b_2+a_{1,1}^2\,b_1+a_{2,2}^2\,b_1\Big)\,\mathrm{D}_1(\phi)\left(y_1, y_2, a_{1,1}, a_{1,2},\right.$$

$$\left.a_{2,2}\right)\Big)$$

$$+\frac{1}{4}\,\frac{1}{a_{1,2}\left(a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2\right)}\left(\sqrt{2}\,\left(a_{2,2}^2-2\,a_{2,2}\,a_{1,1}\right.\right.$$

$$\left.-a_{2,2}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}+a_{1,1}^2+a_{1,1}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}+4\,a_{1,2}^2\right)$$

$$^{1/2}\left(2\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}\;a_{1,2}\,b_2+2\,a_{1,1}\,a_{2,2}\,b_1+\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}\;a_{1,1}\,b_1\right.$$

$$\left.\left.-\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}\;a_{2,2}\,b_1-a_{2,2}^2\,b_1-a_{1,1}^2\,b_1-4\,a_{1,2}^2\,b_1\right)\mathrm{D}_2(\phi)\left(y_1,y_2,a_{1,1},a_{1,2},a_{2,2}\right)\right)$$

$$+\left(\frac{1}{2}\,a_{2,2}+\frac{1}{2}\,a_{1,1}+\frac{1}{2}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}\right)\mathrm{D}_{1,1}(\phi)\left(y_1,y_2,a_{1,1},a_{1,2},a_{2,2}\right)+\left(\frac{1}{2}\,a_{2,2}\right.$$

$$\left.+\frac{1}{2}\,a_{1,1}-\frac{1}{2}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}\right)\mathrm{D}_{2,2}(\phi)\left(y_1,y_2,a_{1,1},a_{1,2},a_{2,2}\right)$$

The functional dependence of ϕ in the above is $\left(y_1,y_2,a_{1,1},a_{1,2},a_{2,2}\right)$ because the $a_{i,j}$ show up as parameters of the transformation. We can supress this dependence via the following substitution:

```
> pde := subs((y[1], y[2], a[1, 1], a[1, 2], a[2, 2])=(y[1],y[2]),pde):
```

This brings the PDE into the following canonical form:

```
> canonical_pde := convert(subs(a=alpha,b=beta,c=mu,x=y,Trace(A.D2) + D1.B + c*phi(vars)),D)
  ;
```

$$canonical\_pde := \alpha_{1,1}\,\mathrm{D}_{1,1}(\phi)\left(y_1,y_2\right)+2\,\alpha_{1,2}\,\mathrm{D}_{1,2}(\phi)\left(y_1,y_2\right)+\alpha_{2,2}\,\mathrm{D}_{2,2}(\phi)\left(y_1,y_2\right)+\beta_1\,\mathrm{D}_1(\phi)\left(y_1,y_2\right) \qquad \textbf{(1.6}$$

$$+\beta_2\,\mathrm{D}_2(\phi)\left(y_1,y_2\right)+\mu\,\phi\left(y_1,y_2\right)$$

The following code finds the explicit forms of the $\alpha_{i,j}$ coefficients:

```
> Vars := [D[1,1](phi)(y[1],y[2]),D[1,2](phi)(y[1],y[2]),D[2,2](phi)(y[1],y[2])];
  for i from 1 to nops(Vars) do:
     eq[i] := factor(rationalize(coeff(canonical_pde,Vars[i]) = coeff(convert(pde,D),Vars[i]
  ))):
  od;
```

$$Vars := \left[\mathrm{D}_{1,1}(\phi)\left(y_1,y_2\right),\mathrm{D}_{1,2}(\phi)\left(y_1,y_2\right),\mathrm{D}_{2,2}(\phi)\left(y_1,y_2\right)\right]$$

$$eq_1 := \alpha_{1,1}=\frac{1}{2}\,a_{2,2}+\frac{1}{2}\,a_{1,1}+\frac{1}{2}\sqrt{a_{2,2}^2-2\,a_{2,2}\,a_{1,1}+a_{1,1}^2+4\,a_{1,2}^2}$$

$$eq_2 := 2\,\alpha_{1,2} = 0$$

$$eq_3 := \alpha_{2,2} = \frac{1}{2}\,a_{2,2} + \frac{1}{2}\,a_{1,1} - \frac{1}{2}\,\sqrt{a_{2,2}^2 - 2\,a_{2,2}\,a_{1,1} + a_{1,1}^2 + 4\,a_{1,2}^2} \qquad \textbf{(1.7}$$

We see that the cross derivative in **canonical_pde** vanishes identically $\alpha_{1,2} = 0$ and the $\alpha_{1,1}$ and $\alpha_{2,2}$ coefficients reduce down to the eigenvalues of $\boldsymbol{A}$:

```
> eq[4] := lambda1 = lambda[1];
  eq[5] := lambda2 = lambda[2];
  eq[6] := isolate(eq[1]-eq[4],alpha[1,1]);
  eq[7] := isolate(eq[3]-eq[5],alpha[2,2]);
  eq[8] := isolate(eq[2],alpha[1,2]);
```

$$eq_4 := \lambda 1 = \frac{1}{2}\,a_{2,2} + \frac{1}{2}\,a_{1,1} + \frac{1}{2}\,\sqrt{a_{2,2}^2 - 2\,a_{2,2}\,a_{1,1} + a_{1,1}^2 + 4\,a_{1,2}^2}$$

$$eq_5 := \lambda 2 = \frac{1}{2}\,a_{2,2} + \frac{1}{2}\,a_{1,1} - \frac{1}{2}\,\sqrt{a_{2,2}^2 - 2\,a_{2,2}\,a_{1,1} + a_{1,1}^2 + 4\,a_{1,2}^2}$$

$$eq_6 := \alpha_{1,1} = \lambda 1$$

$$eq_7 := \alpha_{2,2} = \lambda 2$$

$$eq_8 := \alpha_{1,2} = 0 \qquad \textbf{(1.8}$$

This brings the PDE into the final form:

```
> convert(subs(eq[6],eq[7],eq[8],canonical_pde),diff);
```

$$\lambda 1\left(\frac{\partial^2}{\partial y_1^{\,2}}\,\phi(y_1,y_2)\right) + \lambda 2\left(\frac{\partial^2}{\partial y_2^{\,2}}\,\phi(y_1,y_2)\right) + \beta_1\left(\frac{\partial}{\partial y_1}\,\phi(y_1,y_2)\right) + \beta_2\left(\frac{\partial}{\partial y_2}\,\phi(y_1,y_2)\right) + \mu\,\phi(y_1,y_2) \qquad \textbf{(1.9}$$

This is rather similar to the original form **(1.1)**, except for the lack of the cross derivative. We classify the PDE based on the properties of the eigenvalues of $\boldsymbol{A}$ (this classification scheme applies for all $N \geq 2$):

- **elliptic:** all the eigenvalues are nonzero and have the same sign;
- **hyperbolic:** all the eigenvalues are nonzero and one eigenvalue has a different sign from all of the others;
- **ultrahyperbolic:** all the eigenvalues are non zero and $n > 1$ eigenvalues have different sign from the $N - n > 1$ other eigenvalues; and
- **parabolic:** all the eigenvalues have the same sign except for one, which is zero.

The geometric labelling of the various case come from the similarity between the general PDE and the following quadratic form F (here

written in 2D):

```
> A := 'A':
  B := 'B':
  c := 'c':
  F := X^%T.A.X+B^%T.X+c;
  F := unapply(F,x[1],x[2]):
```

$$F := \begin{bmatrix} x_1 & x_2 \end{bmatrix} . A . \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + B . \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + c \qquad \textbf{(1.10}$$

Here is an example of a parabolic PDE and the a plot of the associated quadratic form $F\left(x_1, x_2\right)$ (the eigenvalues of $\boldsymbol{A}$ are $\lambda_1 = 0$ and $\lambda_2 = 4$):

```
> A := <<2,2>|<2,2>>;
  B := <3,-2>;
  c := 2;
  Eigenvalues(A);
  pde := Trace(A.D2) + D1.B + c*phi(vars);
  plot3d(F(x[1],x[2]),x[1]=-5..5,x[2]=-5..5,axes=boxed,title=expand(F(x[1],x[2])),style=
  patchcontour);
```

$$A := \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$$

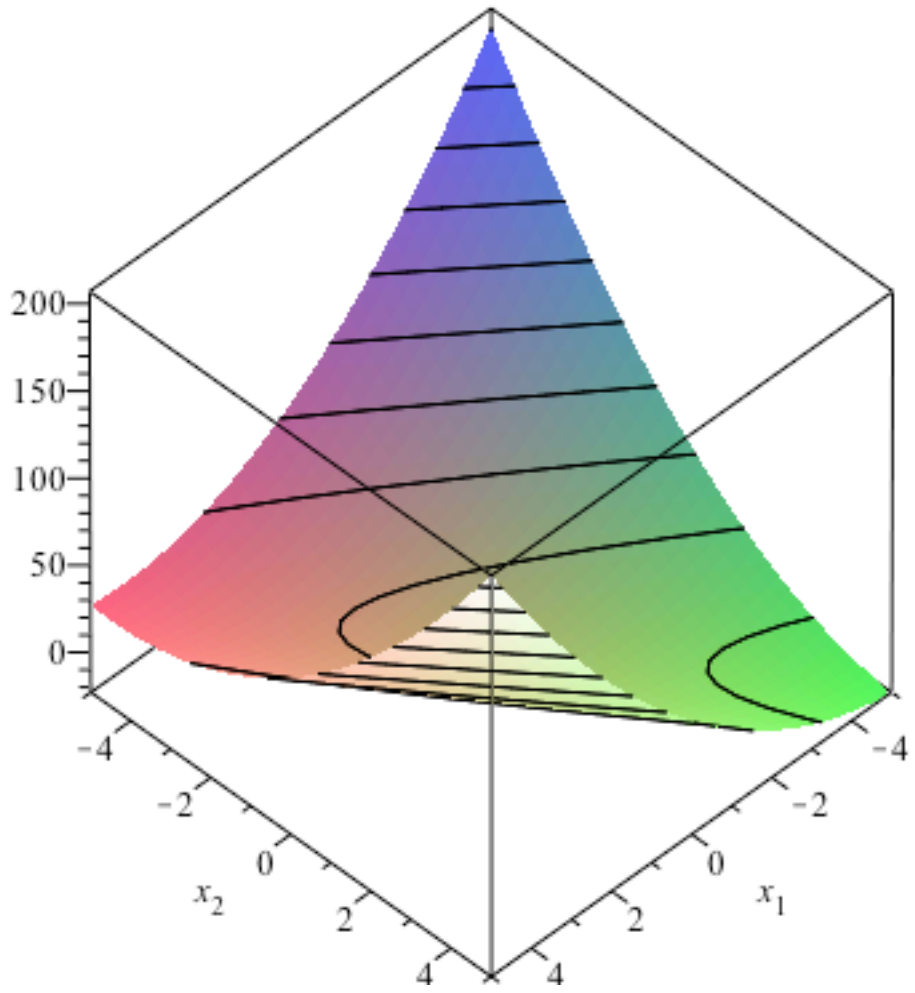
$$B := \begin{bmatrix} 3 \\ -2 \end{bmatrix}$$

$$c := 2$$

$$\begin{bmatrix} 0 \\ 4 \end{bmatrix}$$

$$pde := 2\left(\frac{\partial^2}{\partial {x_1}^2}\,\phi\left(x_1, x_2\right)\right) + 4\left(\frac{\partial^2}{\partial x_2\,\partial x_1}\,\phi\left(x_1, x_2\right)\right) + 2\left(\frac{\partial^2}{\partial {x_2}^2}\,\phi\left(x_1, x_2\right)\right) + 3\left(\frac{\partial}{\partial x_1}\,\phi\left(x_1, x_2\right)\right) - 2\left(\frac{\partial}{\partial x_2}\,\phi\left(x_1, x_2\right)\right) + 2\,\phi\left(x_1, x_2\right)$$

$$2\,x_1^2 + 4\,x_1\,x_2 + 2\,x_2^2 + 3\,x_1 - 2\,x_2 + 2$$



The elliptic case ($\lambda_1 = 3$ and $\lambda_2 = 1$):

```
> A := <<2,-1>|<-1,2>>;
  B := <3,-2>;
  c := 2;
  Eigenvalues(A);
  pde := Trace(A.D2) + D1.B + c*phi(vars);
  plot3d(F(x[1],x[2]),x[1]=-10..10,x[2]=-10..10,axes=boxed,title=expand(F(x[1],x[2])),style=
```

```
patchcontour);
```

$$A := \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

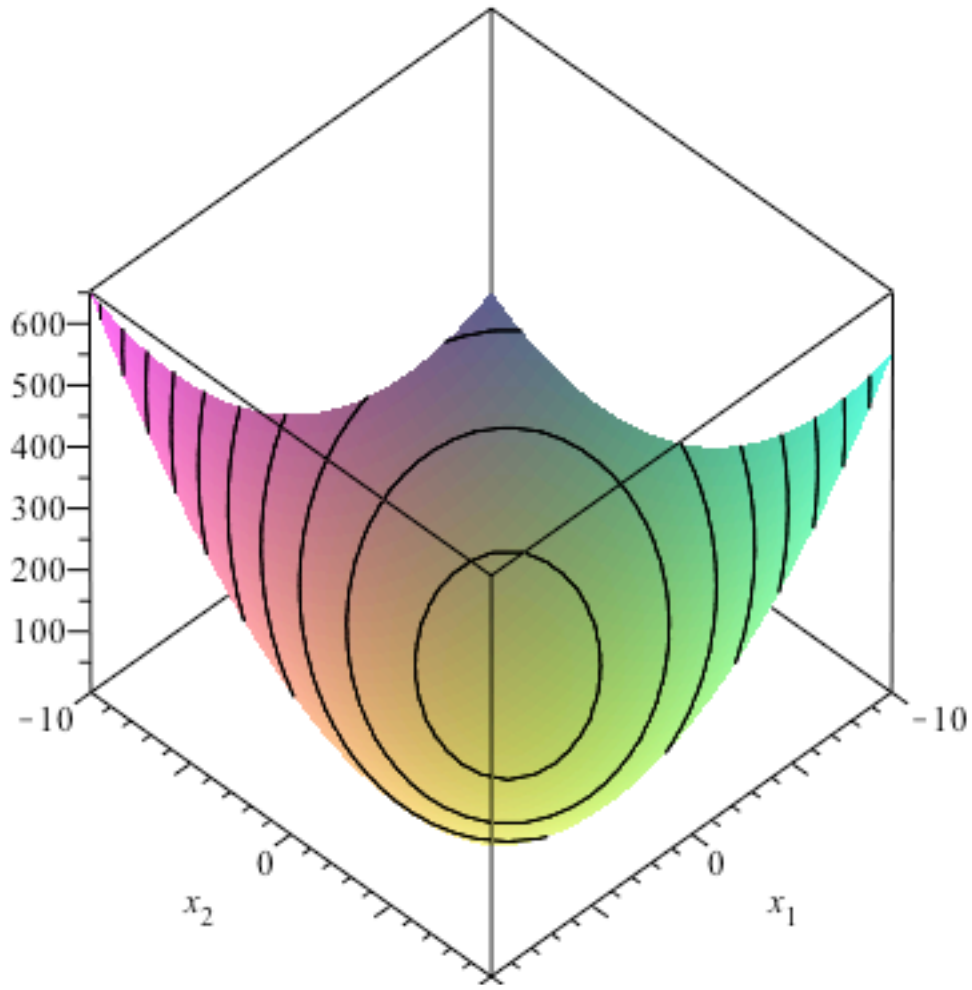
$$B := \begin{bmatrix} 3 \\ -2 \end{bmatrix}$$

$$c := 2$$

$$\begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

$$pde := 2\left(\frac{\partial^2}{\partial x_1{}^2}\,\phi(x_1, x_2)\right) - 2\left(\frac{\partial^2}{\partial x_2\,\partial x_1}\,\phi(x_1, x_2)\right) + 2\left(\frac{\partial^2}{\partial x_2{}^2}\,\phi(x_1, x_2)\right) + 3\left(\frac{\partial}{\partial x_1}\,\phi(x_1, x_2)\right) - 2\left(\frac{\partial}{\partial x_2}\,\phi(x_1, x_2)\right) + 2\,\phi(x_1, x_2)$$

$$2\,x_1^2 - 2\,x_1\,x_2 + 2\,x_2^2 + 3\,x_1 - 2\,x_2 + 2$$



The hyperbolic case ($\lambda_1 = 5\sqrt{2}$ and $\lambda_2 = -5\sqrt{2}$):

```
> A := <<5,-5>|<-5,-5>>;
  B := <3,-2>;
  c := 2;
  Eigenvalues(A);
  pde := Trace(A.D2) + D1.B + c*phi(vars);
  plot3d(F(x[1],x[2]),x[1]=-5..5,x[2]=-5..5,axes=boxed,title=expand(F(x[1],x[2])),style=
```

```
patchcontour);
```

$$A := \begin{bmatrix} 5 & -5 \\ -5 & -5 \end{bmatrix}$$

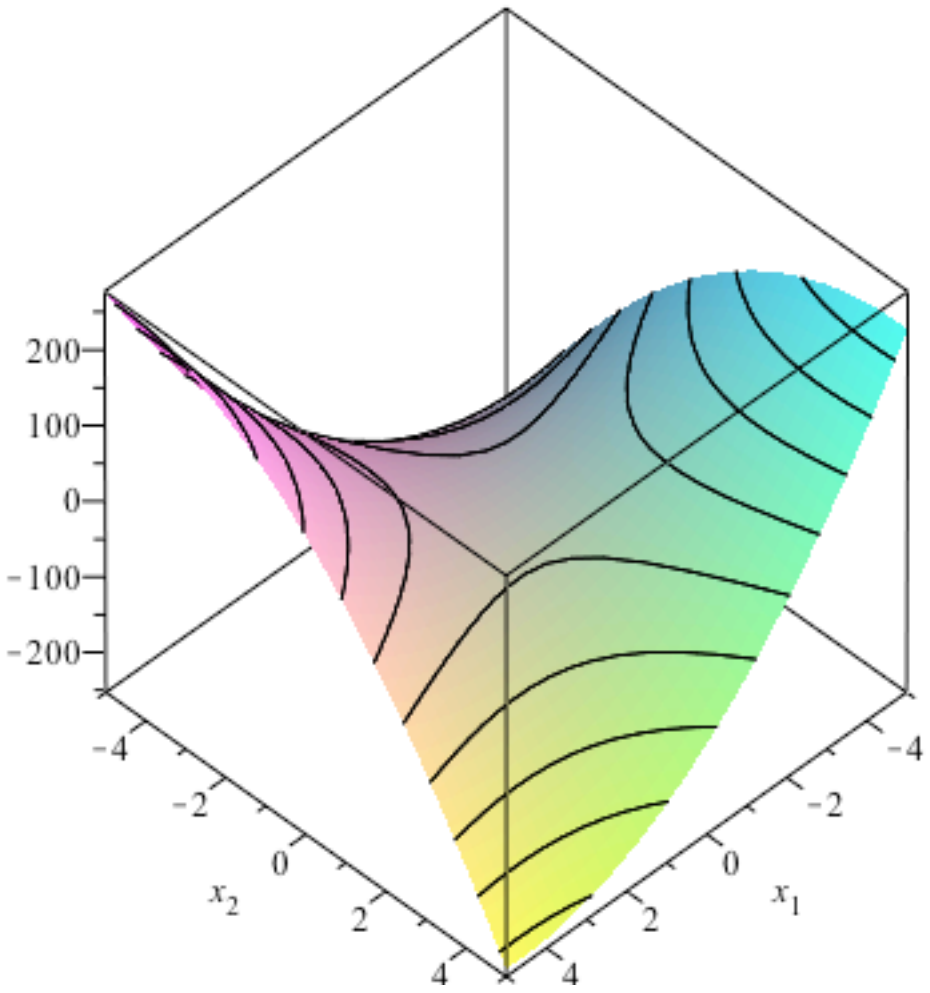
$$B := \begin{bmatrix} 3 \\ -2 \end{bmatrix}$$

$$c := 2$$

$$\begin{bmatrix} 5\sqrt{2} \\ -5\sqrt{2} \end{bmatrix}$$

$$pde := 5\left(\frac{\partial^2}{\partial {x_1}^2}\,\phi(x_1, x_2)\right) - 10\left(\frac{\partial^2}{\partial x_2\,\partial x_1}\,\phi(x_1, x_2)\right) - 5\left(\frac{\partial^2}{\partial {x_2}^2}\,\phi(x_1, x_2)\right) + 3\left(\frac{\partial}{\partial x_1}\,\phi(x_1, x_2)\right) - 2\left(\frac{\partial}{\partial x_2}\,\phi(x_1, x_2)\right) + 2\,\phi(x_1, x_2)$$

$$5\,x_1^2 - 10\,x_1\,x_2 - 5\,x_2^2 + 3\,x_1 - 2\,x_2 + 2$$



Note that there are no ultrahyperbolic PDEs in 2 dimensions.

Analytically solving PDEs using `pdsolve`

```
> restart:
  with(PDEtools):
  with(plots):
```

The following PDEs are classic examples of the different classes of second order equation introduced in the previous section:

```
> parabolic := diff(u(t,x),t)-d*diff(u(t,x),x,x)=0;
  hyperbolic := diff(u(t,x),t,t) - c^2*diff(u(t,x),x,x) = 0;
  elliptic := diff(u(x,y),x,x) + diff(u(x,y),y,y) = 0;
```

$$parabolic := \frac{\partial}{\partial t} u(t,x) - d\left(\frac{\partial^2}{\partial x^2} u(t,x)\right) = 0$$

$$hyperbolic := \frac{\partial^2}{\partial t^2} u(t,x) - c^2\left(\frac{\partial^2}{\partial x^2} u(t,x)\right) = 0$$

$$elliptic := \frac{\partial^2}{\partial x^2} u(x,y) + \frac{\partial^2}{\partial y^2} u(x,y) = 0 \qquad \textbf{(2.1}$$

In addition to these second order equations, there is an important first order equation that appears ofter in the numerical analysis literature; the advection equation:

```
> advection := diff(u(t,x),t) + c*diff(u(t,x),x) = 0;
```

$$advection := \frac{\partial}{\partial t} u(t,x) + c\left(\frac{\partial}{\partial x} u(t,x)\right) = 0 \qquad \textbf{(2.2}$$

Maple can solve each of the above analytically using **pdsolve**:

```
> parabolic_sol := pdsolve(parabolic);
  hyperbolic_sol := pdsolve(hyperbolic);
  elliptic_sol := pdsolve(elliptic);
  advection_sol := pdsolve(advection);
```

$$parabolic\_sol := (u(t,x) = \_F1(t)\,\_F2(x)) \;\&\text{where}\left[\left\{\frac{d}{dt}\,\_F1(t) = \_c_1\,\_F1(t),\ \frac{d^2}{dx^2}\,\_F2(x) = \frac{\_c_1\,\_F2(x)}{d}\right\}\right]$$

$$hyperbolic\_sol := u(t,x) = \_F1(-x - c\,t) + \_F2(x - c\,t)$$

$$elliptic\_sol := u(x,y) = \_F1(y - \mathrm{I}\,x) + \_F2(y + \mathrm{I}\,x)$$

$$advection\_sol := u(t,x) = \_F1(x - c\,t) \qquad \textbf{(2.3}$$

In the last three cases, MAPLE succeeds in obtaining the general solution to the PDE in terms of *arbitrary* functions *_F1* and *_F2*. In the first case, it succeeds in performing a separation of variables and reducing the problem to a pair of ODEs involving a separation constant $\_c_1$ (which we re-label as *k*). We can get it to solve these by using the **build** command:

```
> parabolic_sol := subs(_c[1]=k,u(t,x)=U(t,x,k),simplify(build(parabolic_sol))) assuming
  (k>0);
```

(2.4

$$parabolic\_sol := U(t,x,k) = \_C1\, e^{-\frac{-k\,t\,\sqrt{d}+\sqrt{k}\,x}{\sqrt{d}}}\left(\_C2\, e^{\frac{2\sqrt{k}\,x}{\sqrt{d}}} + \_C3\right) \quad (2.4$$

But this is not the general solution to the PDE, which is given by a superposition of these solutions with different choices of k as follows:

```
> parabolic_general_sol := u(t,x) = int(A(k)*U(t,x,k),k=a..b);
```

$$parabolic\_general\_sol := u(t,x) = \int_a^b A(k)\, U(t,x,k)\, dk \quad (2.5$$

In the above, $A(k)$ is an arbitrary function. Plugging this into the orginal PDE reveals that it is a genuine solution:

```
> gnat := dsubs(parabolic_general_sol,parabolic);
  gnat := combine(dsubs(parabolic_sol,gnat));
```

$$gnat := \int_a^b A(k)\left(\frac{\partial}{\partial t} U(t,x,k)\right) dk - d\left(\int_a^b A(k)\left(\frac{\partial^2}{\partial x^2} U(t,x,k)\right) dk\right) = 0$$

$$gnat := 0 = 0 \quad (2.6$$

Alternatively, we can use the pdetest command to check that **parabolic_general_sol** solves the parabolic PDE:

```
> pdetest(subs(parabolic_sol,parabolic_general_sol),parabolic);
```

$$0 \quad (2.7$$

The moral of the story is that **pdsolve** by itself does not always return the general solution of a differential equation, unlike **dsolve**. Also, one quickly discovers that it fails to return solutions for many different PDEs of interest. This reflects the basic truth that PDEs are harder to solve than ODEs.

Numerically solving PDEs using pdsolve/numeric

Let's now turn our attention to the following wave equation:

```
> V := 'V':
  pde := diff(phi(t,x),t,t) - diff(phi(t,x),x,x) + V(x)*phi(t,x);
```

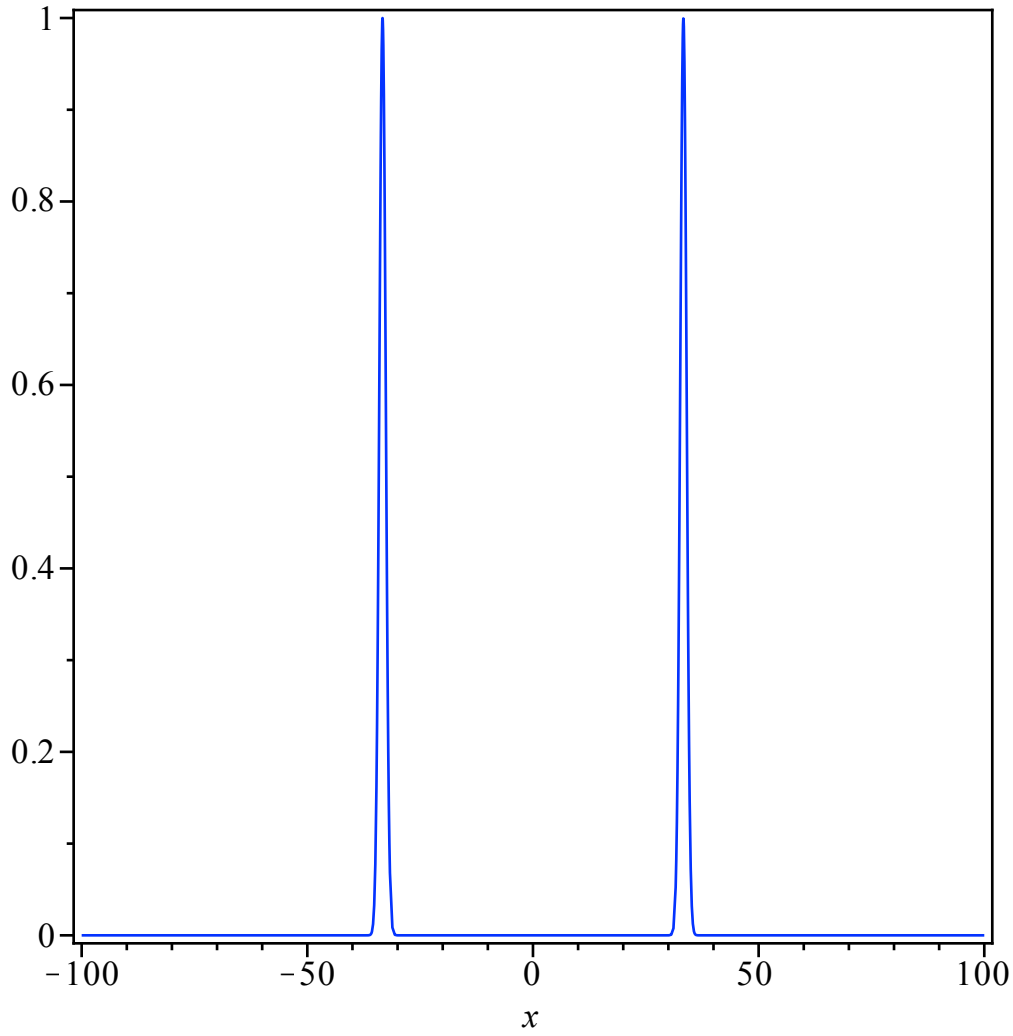
$$pde := \frac{\partial^2}{\partial t^2}\phi(t,x) - \left(\frac{\partial^2}{\partial x^2}\phi(t,x)\right) + V(x)\,\phi(t,x) \quad (3.1$$

Here, $V(x)$ is a potential function. We choose a potential as follows:

```
> x0 := 100;
  V := x -> exp(-(x-x0/3)^2) + exp(-(x+x0/3)^2);
  p1 := plot(V(x),x=-x0..x0,axes=boxed,color=blue):
  p1;
```

$$x0 := 100$$

$$V := x \to e^{-\left(x - \frac{1}{3} x0\right)^2} + e^{-\left(x + \frac{1}{3} x0\right)^2}$$



In this case, pdsolve can succeed in separating variables, but cannot integrate the resulting set of ODEs:

```
> build(pdsolve(pde));
```

$$\phi(t,x) = e^{\sqrt{-c_1}\,t}\, DESol\left(\left\{\frac{d^2}{dx^2}\_Y(x) - \_Y(x)\,\_c_1 - \_Y(x)\,e^{-\frac{1}{9}(3x-100)^2} - \_Y(x)\,e^{-\frac{1}{9}(3x+100)^2}\right\}, \{\_Y(x)\}\right)\_C1 \quad \textbf{(3.2}$$
$$+ \frac{DESol\left(\left\{\frac{d^2}{dx^2}\_Y(x) - \_Y(x)\,\_c_1 - \_Y(x)\,e^{-\frac{1}{9}(3x-100)^2} - \_Y(x)\,e^{-\frac{1}{9}(3x+100)^2}\right\}, \{\_Y(x)\}\right)\_C2}{e^{\sqrt{-c_1}\,t}}$$

So we instead look for a numeric solution. To do so, we need to choose initial and boundary conditions for $u(t,x)$, which we arrange in a list **IBC**:

```
> f := x -> exp(-(x-x0/2)^2/16);
  IBC := [phi(0,x)=f(x),D[1](phi)(0,x)=D(f)(x),phi(t,-2*x0)=0,phi(t,+2*x0)=0];
```

$$f := x \to e^{-\frac{1}{16}\left(x - \frac{1}{2}x0\right)^2}$$

$$IBC := \left[\phi(0,x) = e^{-\frac{1}{16}(x-50)^2}, D_1(\phi)(0,x) = \left(-\frac{1}{8}x + \frac{25}{4}\right)e^{-\frac{1}{16}(x-50)^2}, \phi(t,-200) = 0, \phi(t,200) = 0\right] \quad \textbf{(3.3}$$

The solution module is obtained by calling **pdsolve** with the option **numeric**. Note the **timestep** and **spacestep** command are optional.

```
> pde_sol := pdsolve(pde,IBC,numeric,timestep=1/4,spacestep=1/2);
```

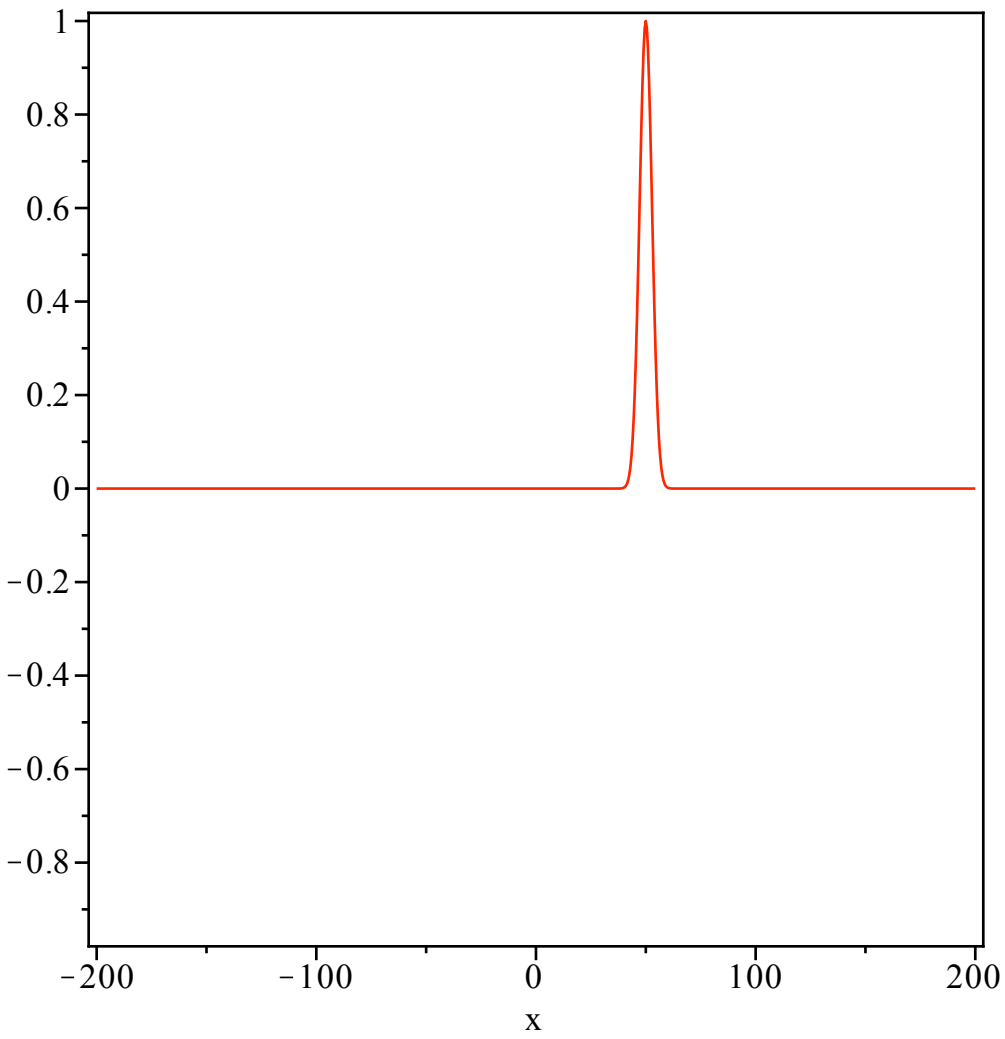
$$pde\_sol := \textbf{module}(\,)\ \textbf{export}\ plot, plot3d, animate, value, settings;\ \ldots\ \textbf{end module} \quad \textbf{(3.4}$$

We can generate a movie of the solution by calling the following:

```
> movie := pde_sol:-animate(t=0..250,axes=boxed,frames=50):
```

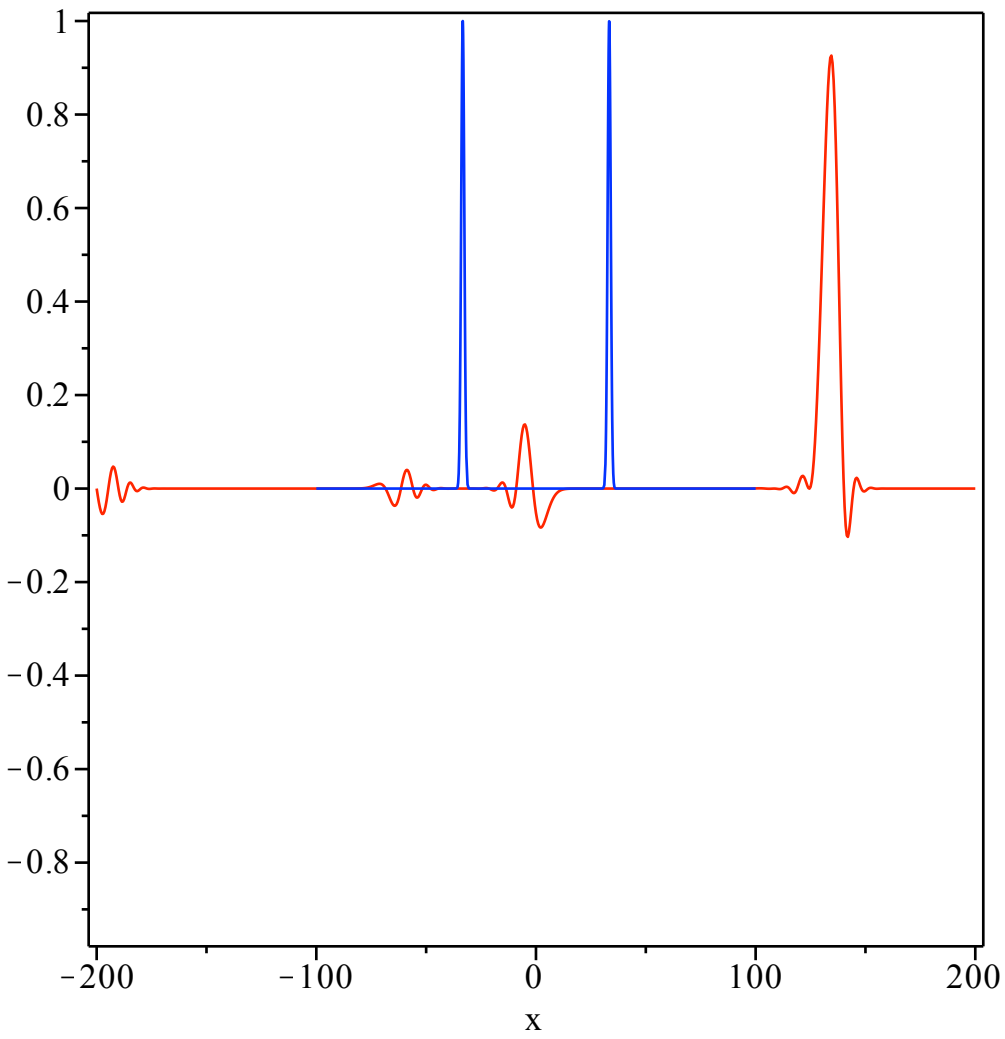
Notice in the above, instead of displaying the movie, we've assigned it to a variable. We can get Maple to diplay the actual movie by just calling **movie** (click on it to play).

```
> movie;
```



We can also superimpose the potential on the movie by using display:

```
> display([movie,p1]);
```



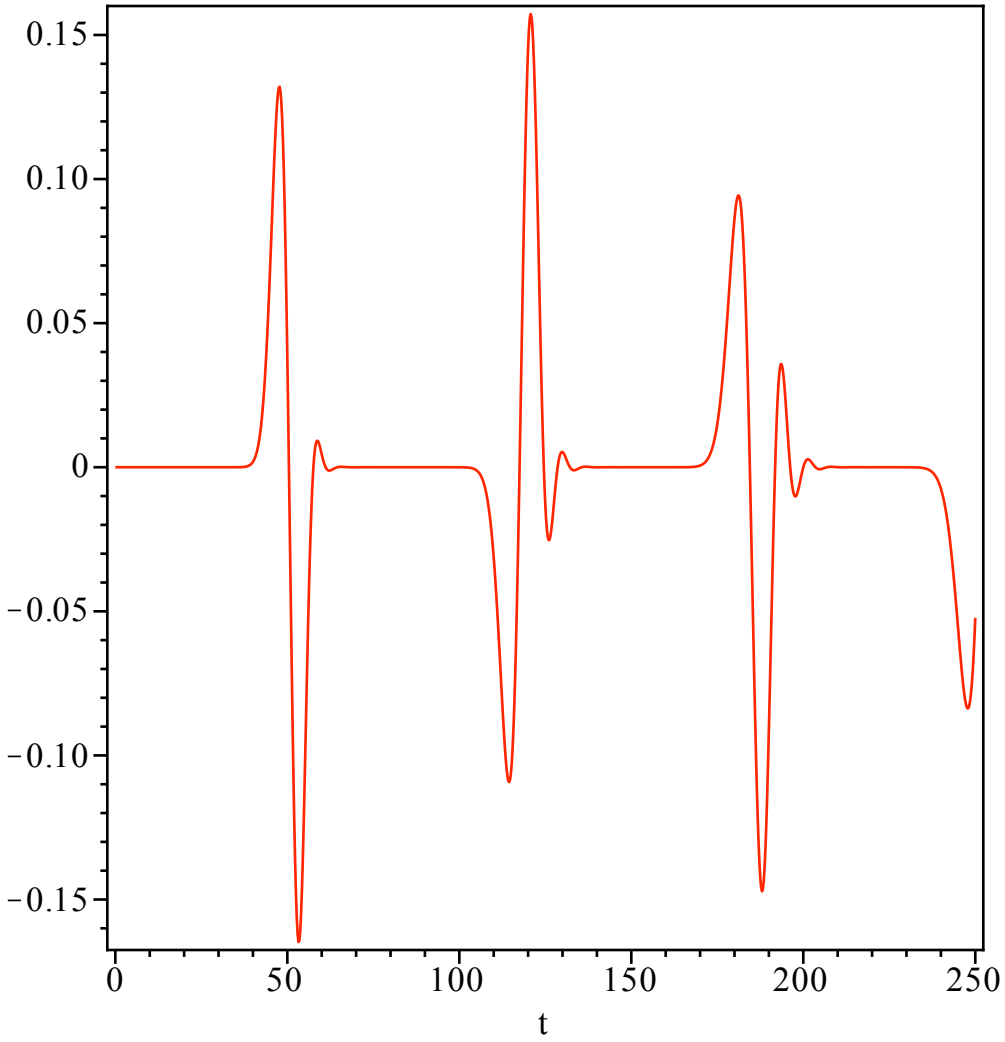
There are a variety of other ways to interact with **pdsolve/numeric**. For example the following generates a 3D plot of $u(t, x)$ over a given range in t and x:

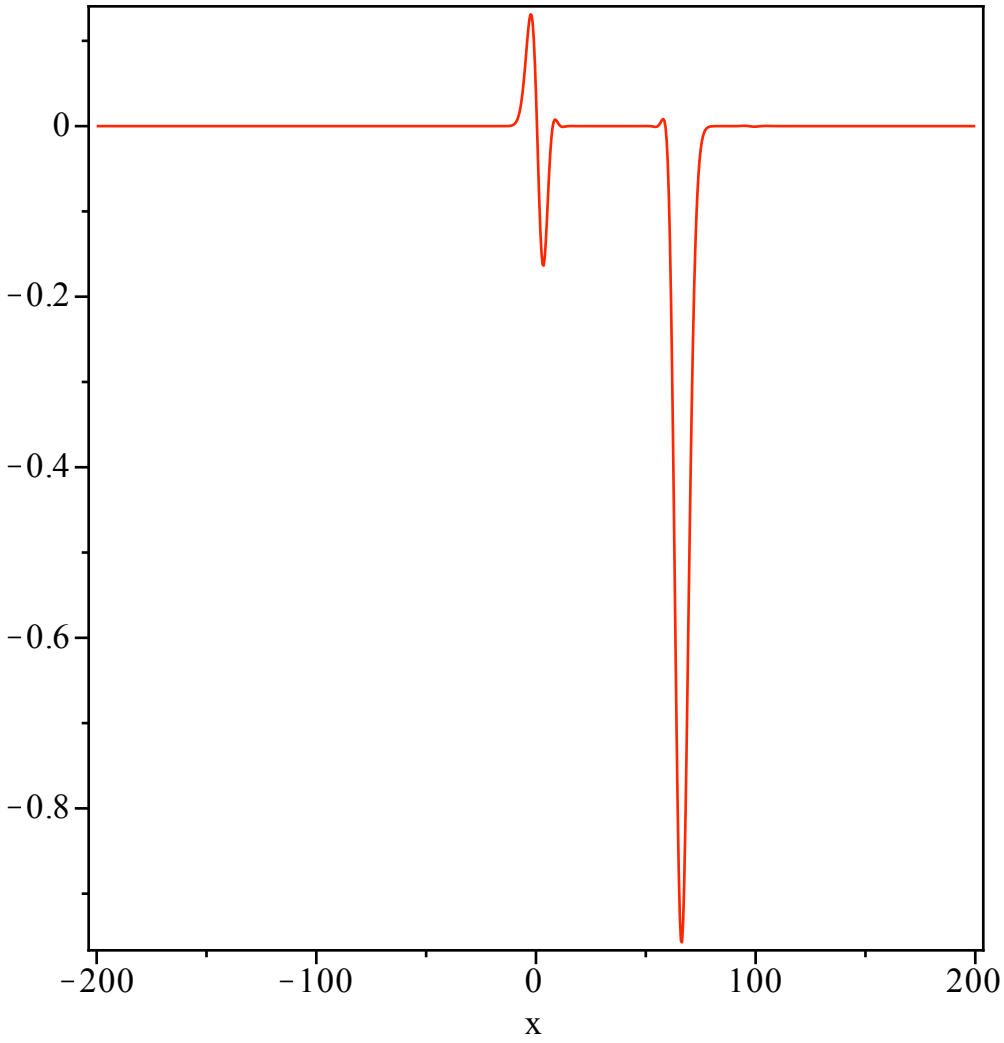
```
> p2 := pde_sol:-plot3d(t=0..200,x=-x0..x0,grid=[100,100],style=patchnogrid,axes=boxed,
  shading=ZHUE):
> p2;
```

p2 **(3.5**

Finally, we can just plot u(t,x) as a function of t for a fixed value of x, or vice versa:

```
> pde_sol:-plot(x=0,t=0..250,axes=boxed);
  pde_sol:-plot(t=50,x=-2*x0..2*x0,axes=boxed);
```




0
-0.2
-0.4
-0.6
-0.8
-200
-100
0
100
200
x